National College of Ireland

Comparative Analysis of Malware Investigative Tools

MSc Research Project
MSc Cyber Security

Ian Ngugi Wamunyu
Student ID: x20110448

School of Computing
National College of Ireland

Supervisor: Dr Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Ian Ngugi Wamunyu……

Student ID: ……X20110448……..……

Programme: …… MSc Cyber Security…… **Year:** ……2023……..

Module: …… Research Project ……..……

Supervisor: ……Dr Imran Khan……..……

Submission Due Date: ……18/09/2023……..……

Project Title: …… Comparative Analysis of Malware Investigative Tools…

Word Count: …5719…… **Page Count**……25……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Ian Ngugi Wamunyu……

Date: ……18/09/2023……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Malware Investigation Tools

Ian Ngugi Wamunyu
X20110448

Abstract

Malware is a multipurpose attacking software that can be used in a variety of cyber-attacks, from encrypting a government institution's data using ransomware to installing adware on the local cyber-cafe. Due to its high-risk factor and harm to businesses and individuals, multiple malware analysis tools are available to investigate malicious software. This paper focuses on comparing different tools for analysing malware across the different malware analysis types i.e., Static, dynamic/behavioural, code and memory analysis. The objective of this research is to enable cyber and malware analysts a detailed reference of tools in terms of accuracy, ease of use, community support and most importantly the tools' analysis capability.

Organisations such as the Health Service Executive (HSE) of Ireland and Medibank an Australian health insurance company may benefit from this comparative analysis given their recent malware attack in 2021 and 2022 respectively.

Results

Keywords (Malware, Malware Analysis, Cybersecurity)

1 Introduction

Cyber incidents are on the rise with many services and businesses getting automated due to Covid 19. The epidemic created an opportunity for businesses to cut costs on renting building spaces, electricity, heating etc due to employees working from home for almost 2 years. This automation led to an increase in cyber-attacks using malicious software programs commonly known as malware.

In 2021, the Health Service Executive (HSE) of the Republic of Ireland fell to a ransomware attack causing alter discord to a majority of hospitals and clinics in Ireland, affecting patient records, appointment bookings and email systems, causing them to shut down most of the I.T. systems and go back to using paper-based systems(Reevell, 2021). The attackers demanded a ransom if the institution wanted their data back, in many cases companies would pay the ransom but would not get the data back. (IBM, Security, 2022) IBM's report on the cost of data breaches illustrates that the health services industry is one of the most targeted industries with an average of $10 million breach cost.

In 2021, the Colonial Pipeline- one of the U.S.A.'s largest pipelines providing almost half of the east coast's fuel ranging from diesel, home-heating oil to military fuel came to a halt after a malware attack which took their computer systems offline for several days(Charlie Osborne, n.d.).Cybercrime magazine predicted that the damage cost of ransomware would rise from approximately $320 million in 2015 to over $20 billion in 2021(Freeze, 2018). The severe damage that malware programmes cause leads to the significance of this research into the most efficient ways to use malware analysis tools to detect and prevent such attacks (Afianian et al., 2019).

A recent malware attack is the Medibank ransomware attack that occurred in October 2022. Being one of Australia's leading private health insurers, the company incurred stolen data for over 9 million past and current clients with approximately 15%-20% being international customers. The hackers stole client names, addresses, mobile numbers and medical history and demanded a 15 million Australian dollar ransom(Whiteman, 2022).

This has led to the question; Which is the most efficient way malware analysts can choose malware analysis tools to analyse malware programs in the context of current business and industrial practices?
The proposed research question is worth investigating given the rising financial and software damages caused by malware attacks. The proposed approach will enable malware analysts and cybersecurity experts to detect, analyse and prevent future malware attacks beforehand or have an effective countermeasure in the occurrence of a malware attack. The research question is feasible in terms of a variety of open-source malware analysis tools. The researcher will choose different malware analysis tools depending on measurables such as the tool's rating, the specific sector of malware it targets and the range of the analysis of the tool. The analysis will be conducted on a virtual machine- a separate and safe environment to test malicious programs which will block the spread of malware in the occurrence of a compromised system (Sikorski, M and Honig, A, 2012)
The rest of the paper is divided into sections, section 2 is the review of past/recent works of literature, covering the description of malware, types of malware, the malware kill chain and malware analysis. Additionally, a review of recent papers covering different malware analysis tools is conducted and a comparison table is created. Section 3 covers the Research methodology, section 4 is the design specifications, section 5 shows the Implementation of the research, Section 6 focuses on the project evaluation and the results of the analysis, section 7 is the conclusion and future work section 8 is the references.

2 Related Work

This section explains what malware is, malware types, the malware kill chain and malware analysis. The second part reviews related work focusing on different malware analysis tools with a comparison table.
Malware is a general term used to describe a malicious program/code. This program can be used to cause harm/damage to computer systems, networks, and hardware devices.
(Greene, 2004; Malwarebytes, n.d.) defines malware as the cause of detrimental computing behaviour via a weakness in the device triggered by a software program.

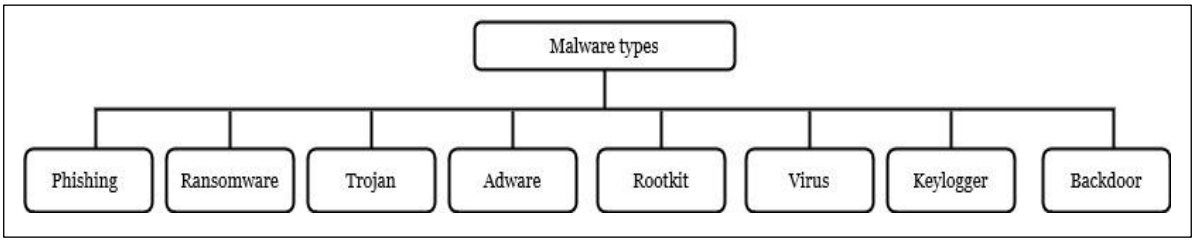

Fig. 1. Malware-type diagram

As illustrated in Fig 1, malware range from social engineering-enabled email attacks, which trick an individual into clicking a malicious link via email or text commonly known as phishing. There is also ransomware- where malicious programs that can lock a victim out of their computer, encrypt their files, and later demand payment to decrypt the victim's files(Egele *et al.*, 2008). Trojan is malware that masks itself as normal software to trick a user into installing it, this enables the trojan to acquire sensitive information, monitor user activity or send phishing emails to the user's contact list. Trojan malware can also infect a computing device with software that pops up unwanted advertisements commonly known as adware(KA, 2018). Rootkit is malicious software that gains privileged access to a target computing device

and hides itself until commanded to attack. A virus is harmful software that copies itself and can infect other computing devices. Keyloggers are malware that monitors a victim's keystrokes and can send them to a cyber attacker. Backdoor also known as RAT (remote access Trojan) is a malware that allows an intruder to gain administrator access and perform commands on the target's device.

2.1 Malware Analysis

Malware analysis is the scientific study of how malicious software reacts(K.A, no date). This procedure enables cyber security and malware analysts the ability to:

- find out the makeup and function of malicious software.
- conduct a system audit of how it was made vulnerable and to which extent.
- use the findings to compare similarities with new malware regarding network markers (IP addresses), filenames and registration keys.
- better understanding the target of the attack by the type of malware used i.e., getting keystrokes using a keylogger can point to stealing passwords.

Fig.2 presents the 4 main types of malware analysis.

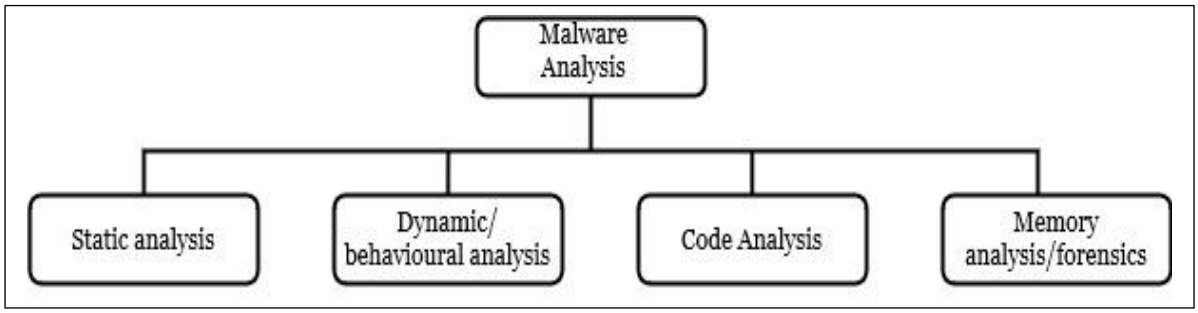

Fig. 2. Malware Analysis diagram

Malware analysis is many divided into:

- Static analysis

Like penetration testing's white box analysis, the malicious file is examined without executing it in a computing device. It's a primary analysis procedure that enables an analysis to correctly classify using the information retrieved.
In this section, the malware analyst focuses on a file's hashes, this is a unique fixed-length strings obtained from hashing algorithms such as SHA(Secure Hash Algorithm) or MD5 (Message Digest 5)(Barker, 2021).

- Dynamic/ behavioural analysis

This is the procedure of running a suspicious sample/ file in a controlled space commonly known as a sandbox. A sandbox is an isolated environment on a computing device, where a malware analyst observes the malware's real-time behaviour. When conducting a dynamic analysis, the malware analyst will revert to the virtual machine's previous snapshot, launch the dynamic analysis tools, execute malware samples using administrative authority and finally

analyse the report from the analysis tools. The report will enable the classification of the sample from its behaviour and functioning.
In this section, the malware analyst monitors the malware process tree- focusing on child processes from the portable executable, network traffic, process, and memory performances. Dynamic analysis is also useful in validating static analysis results(KA, 2018).

- Code Analysis

This type of analysis focuses on investigating the sample's code. It is further divided into static code analysis and dynamic code analysis. Both subsections are like static and dynamic analysis but their main focus is on the sample code(Fortinet, 2023).

- Memory analysis/ forensics

This procedure is especially useful in pointing out how sneaky and evasive the malicious programs are. The technique observes the behaviour of the malware after infecting a computing device by studying the computer's (RAM) Random Access Memory(Dener, Ok and Orman, 2022).

2.2 Malware attack life cycle

As depicted in Fig. 3, malicious software goes through 4 major stages to attack a target device, the malware is developed by a cyber attacker usually with technical knowledge of how computing devices work and software coding experience. A 'script kiddie'- is an inexperienced malicious hacker, who can get pre-developed malware samples from the dark web. The malware is distributed depending on the type of malware or type of attack. For example, a USB drive can be left in the reception of a business where the receptionist can try to use it on the company's computer. An attacker can also send an employee a phishing email with an executable file. Once the file is clicked on, the malware infects the target's device/ network. At this stage depending on the type, the malware can either inject the computing device with advert pop-ups, monitor the target's keystroke or conceal itself until it is issued an order by its commander.

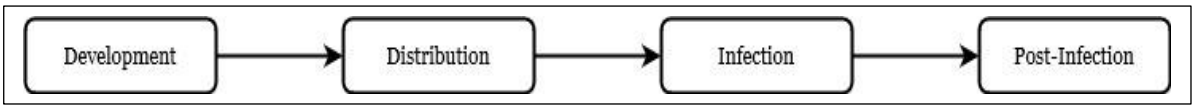

Fig 3. Malware Attack life cycle(Mohanta and Saldanha, no date)

2.3 Static Analysis

(Sihwail et al., 2018) focuses on surveying the different malware analysis techniques. The researcher collects various survey papers on malware, malware types and malware analysis and categorises them based on their field of research and results. A noticeable difference between this research and the proposed research is the practical and real-time application of malware analysis tools.

(Yousuf *et al.*, 2023) look more in-depth at detecting malware using static analysis in the Windows platform. Focusing on collecting various Windows portable executable malware and categorising them based on dynamic-linked-library (DLLs), application programming interface (API) functions, PE headers and sections.

(Balodi *et al.*, 2023) this paper mainly covers static analysis attributes and studies the results using a machine learning model commonly known as Random Forest. This research analysis is part of the malware analysis subsection where else we will be looking at all four subsections which are static, dynamic, code and memory analysis.

2.4 Dynamic Analysis

(Maniriho et al., 2022) The research mainly focused on behavioural/dynamic analysis. Providing a detailed approach to the analysis steps and different dynamic analysis tools. (Egele et al., 2008). focuses on the same malware analysis technique but conducts an extensive survey on the technique with different types of behavioural analysis tools. (OR-MEIR et al., 2020) also focuses on dynamic analysis by conducting a survey. The researcher focuses more on the analysis flow dividing it into live OS, volatile memory forensic and side-channel analysis.

(Afianian *et al.*, 2020) focuses on comparing manual and automated dynamic analysis avoidance procedures of malware. The researcher grouped the analysis into anti-debugger for manual analysis evasion and sandbox evasion for automated analysis evasion. This research focuses mainly on how malware evades analysis tools whereas we focused on the different attributes of different analysis tools.

2.5 Tools and executables

(Preeti and Agrawal, 2022) analysis of three different analysis tools namely Cuckoo sandbox, any-run and integer analysis. (Ilić *et al.*, 2022) also focused on the Cuckoo sandbox tool and the Drakvuf sandbox. They both check-listed the analysis tools on different features such as scalability, reporting, execution time and signatures.

(Shijo and Salim, 2015) focuses on combining static and dynamic analysis for the detection of malware. Their integrated model scores almost 99% accuracy rating as compared to the roughly 95% and 97% in dynamic and static methods respectively.

(Hampton, Baig and Zeadally, 2018) the research focuses on the malicious exploit known as ransomware and how it acts in Windows operation systems. The Researcher analysis multiple ransomware strains against the API and compares the results to the normal state of a Windows platform. This research also illustrates the rise of ransomware and influenced the use of this executable as our malicious program.

2.6 Research Niche

The related work focuses on different aspects of malware, malware analysis and malware investigative tools. This research is different from related works due to the application of all major malware analysis types; static, dynamic, code and memory analysis. This distinction will enable malware analysis the ability to efficiently focus on a set of malware tools depending on their requirements.

	PAST PAPERS	AREA FOCUSED
1	(Aslan and Samet, 2017)	Malware analysis tools
2	(Lebbie, Prabhu and Agrawal, 2022)	Dynamic analysis tools
3	(Kayani and Saeed, 2021)	Anti-virus evasion
4	(Madan, Sofat and Bansal, 2022)	tools
5	(Preeti and Agrawal, 2022)	Malware analysis tools
6	(Ilić *et al.*, 2022)	Malware analysis tools
7	(Maniriho, Mahmood and Chowdhury, 2022)	Dynamic analysis
8	(Egele *et al.*, 2008)	Dynamic analysis
9	(Sihwail, Omar and Ariffin, 2018)	Static analysis
10	(Shijo and Salim, 2015)	Static & dynamic analysis
11	(Afianian *et al.*, 2020)	Dynamic Analysis evasion
12	(Yousuf *et al.*, 2023)	Static analysis
13	(Balodi *et al.*, 2023)	Static analysis/ machine learning
14	(Hampton, Baig and Zeadally, 2018)	Ransomware exploit

Table 1. Previous papers Review

3 Research Methodology

This section covers the methodology used to carry out the malware analysis process. Section 3.1 looks at the research method which involves the gathering of a portable executable, virtual environment setup, malware execution, malware analysis, presentation of results and the comparative analysis of the different investigation tools. Section 3.2 covers research requirements including dataset information, hardware, and software requirements. Section 3.3 covers the ethical considerations needed in this research.

3.1 Research Method

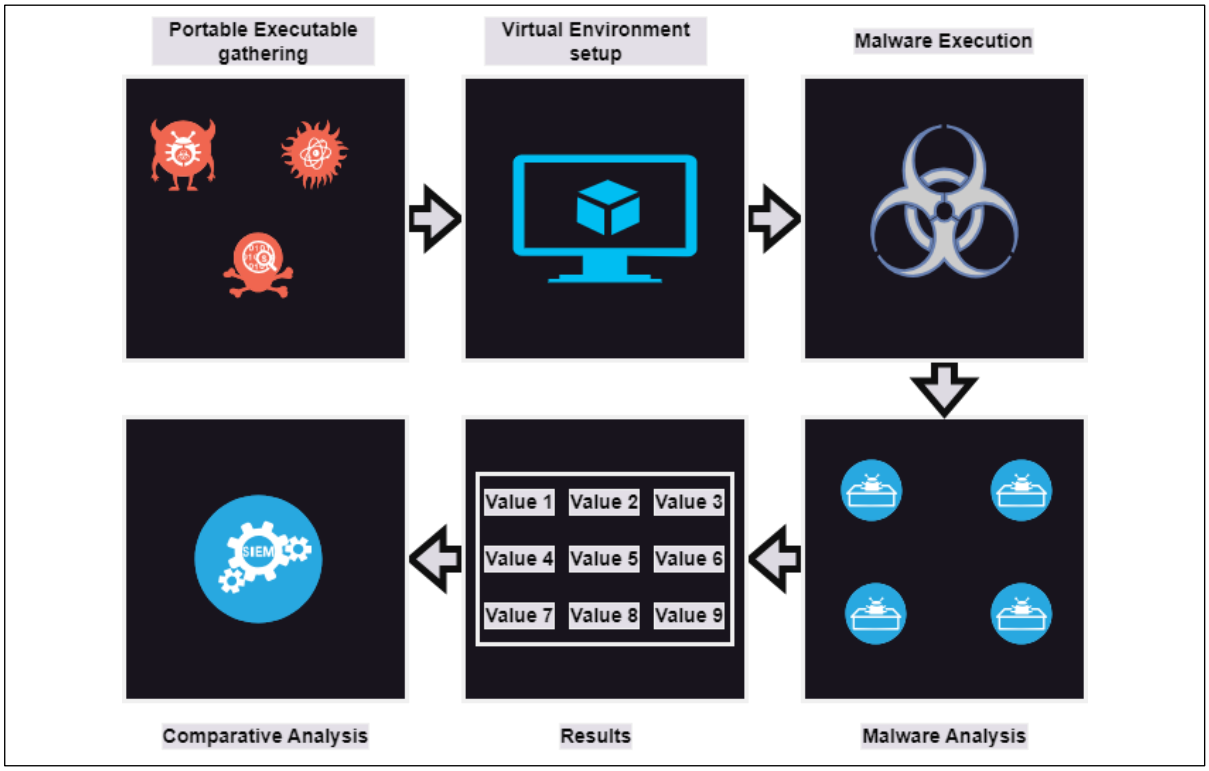

Fig. 4. Research Methodology

3.1.1 Portable Executable Gathering

Malware samples are provided via theZoo (Nativ and Shalev, no date): A live repository for malicious software. This database contains open and public malware dataset(s) for malware analysis. The database was created by Yuval Tisf Nativ and is looked after by Shahak Shalev. These samples are only for research and study purposes.

This research focuses on the ransomware portable executable known as WannaCry. The dataset comes compressed and password protected. Measures were taken to handle the executable in a controlled environment without real-time access to different devices networks or the internet.

3.1.2 Virtual Environment setup

The analysis is set up in Virtual-box(*Oracle VM VirtualBox*, no date) virtual machine, which is a safe and controlled environment for malware analysis by Oracle. In the likely event that

the malware sample may corrupt or infect the machine, the platform enables the revert to an earlier snapshot before the malware is launched.

3.1.3 Malware Execution

This process is divided into two categories where the executable is analysed without running it while doing static analysis. The other analysis requires the execution of the malware.

3.1.4 Malware analysis

This stage begins with the identification and installation of multiple tools ranging from static, dynamic, code and memory analysis tools. After the installation process, a snapshot of the virtual machine is saved before the execution of the malware. Once a tool is used and the malware is executed, the virtual machine reverts to the previous 'clean'-uncorrupted snapshot and the process is repeated with another analysis tool.

3.1.5 Results

After each malware analysis tool is used, the results are saved either in a text file or a screenshot.

3.1.6 Comparative analysis

This is the final stage where all the documented results from the analysis are examined and compared focusing on the given deliverables.

- User/ Community Support: tools support when a user encounters an issue.
- Features: Different attributes that may benefit the analysis.
- Usability: the ease of use of the tool.
- Efficiency: Comparing the tool's main functions and results
- Cost: Internal and external cost parameters of the tool

3.2 Research Resources

The research resources consist of hardware and software categories.

3.2.1 Hardware resources:

Host machine	
Computer	HP
RAM	16GB
Memory	1TB
Operating system	Windows 11
Virtual Machine (Virtual Box)	
Operating system	Windows 10
RAM	8GB
Memory	80GB

Table 2. Hardware resources

3.2.2 Software resources:

Malware Analysis Tools

Name	**Category**	**Program**	**Operating System**
Virtual box	Virtual machine	Open Source	Cross-platform
Ninite (multiple basic system applications		Open Source	Windows
Virus Total	Static tools	Open Source	Cross-platform
PEStudio		Open Source	Windows
ProMonitor	Dynamic tools	Open Source	Windows
Regshot		Open Source	Windows
Volatility	Memory tools	Open Source	Cross-platform
GRR Rapid Response		Open Source	Cross-platform
Radare2	Code tools	Open Source	Cross-platform
Ghidra		Open Source	Cross-platform

Table 3. Software resources

- Virus Total(virus total, 2023): this is an online analysis tool used to scan documents/ libraries and uniform resource locators (URLs) for malware. The service compares the file hashes (signature) against previously detected malware in their database. Most malware analysis tools have a Virus Total link that uses their online services.
- PEStudio (Ochsenmeier, 2023): A Windows application for analyzing and investigating portable executables (PE).The application searches for dynamic link libraries (DLLs) and executable binaries.
- ProMonitor(Russinovich, 2023): Similar to Windows's task manager, Process monitor displays real-time system information which includes registry, process, memory and network activities. Developed by Microsoft and is part of the Sysinternals suite.
- Regshot(TiANWEi, 2023): an open-source file and registry analysis tool, that compares pre and post-infection snapshots.
- GRR Rapid Response ('google/grr', 2023): This is a cross-platform remote memory analysis tool that uses YARA libraries.
- Volatility (Volatility Foundation, 2020): Developed by the Volatility Foundation, this framework is used in digital forensics, it monitors a system's random access memory (RAM) and obtains more information via memory dumps.
- Radare2 (*radare*, no date): This tool can assemble and disassemble malware files and monitors the components on the NoSQL database. Compatible with Android, Solaris, Windows, Linux and macOS operating systems.
- Ghidra (*Ghidra*, no date): created and maintained by the National Security Agency Research Directorate, this tool enables analysts to examine malware code and re-engineer it. This helps the analyst to understand the exploit on a more detailed level. Compatible with Windows, Linux and macOS operating systems.

3.3 Ethical Considerations of the Research

The research mainly focuses on setting up a controlled environment in a computing device not connected to any institution or company network .in case the malware sample can break through, the researcher can backtrack to a previous snapshot of the virtual machine and format the infected one. The malware samples are available.

- GitHub on the Zoo: A live Malware Repository. Contains an open and public malware dataset(s) for malware analysis.

The samples are for research purposes only.

4 Design Specification

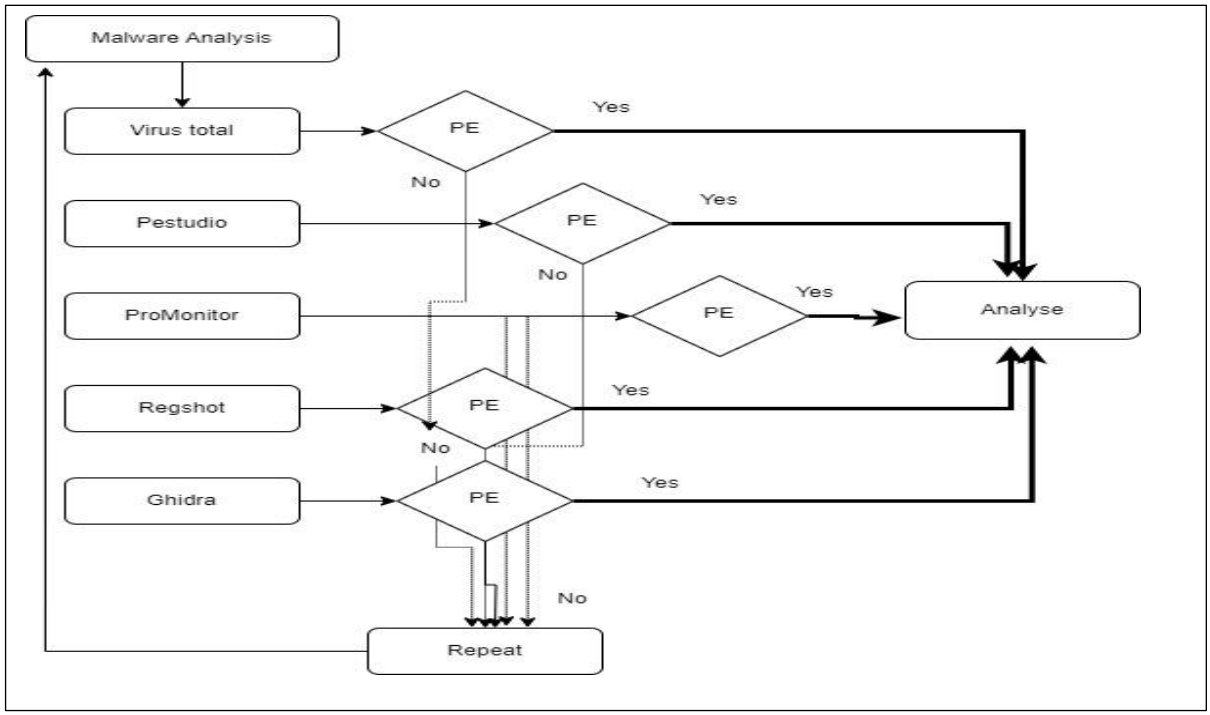

Fig. 5. Research flowchart

Fig 5 shows a detailed view of the malware analysis process where the (PE) Portable executable is examined by the analysis tool. Once the PE is flagged as a malicious program, all the data is taken for comparative analysis.

5 Implementation

The research was implemented in VirtualBox -a virtual machine to enable the safe handling of the malware and revert to an uninfected state after analysing the malware with a set of tools.

Fig. 6. Oracle VM VirtualBox Manager

Figure 7 shows the VirtualBox sandbox setup: Operating system-windows 10 (64bit), memory-8GB, processors -2, graphics controller-VBoxSVGA, storage-80GB, shared folders disabled, network-adapter 1(NAT network) and(not-attached) while conducting analysis.

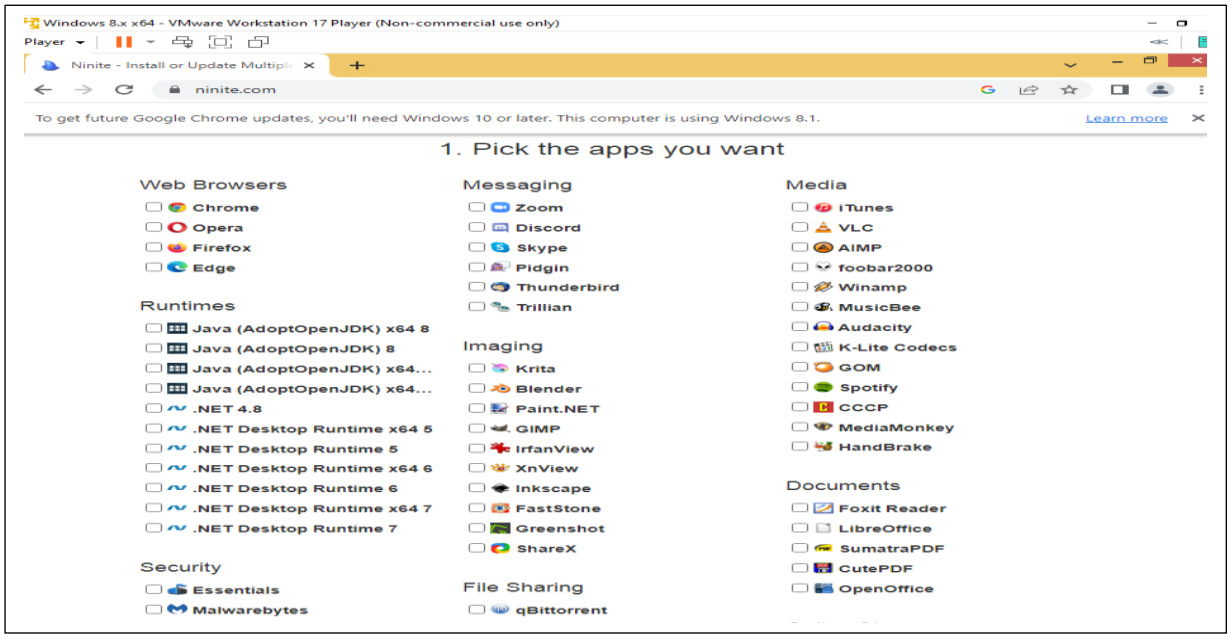

Fig. 7. Basic application setups

Fig8 shows the use of ninite software to download multiple basic applications to mimic a normal computing environment(Swieskowski and Kuzins, no date).

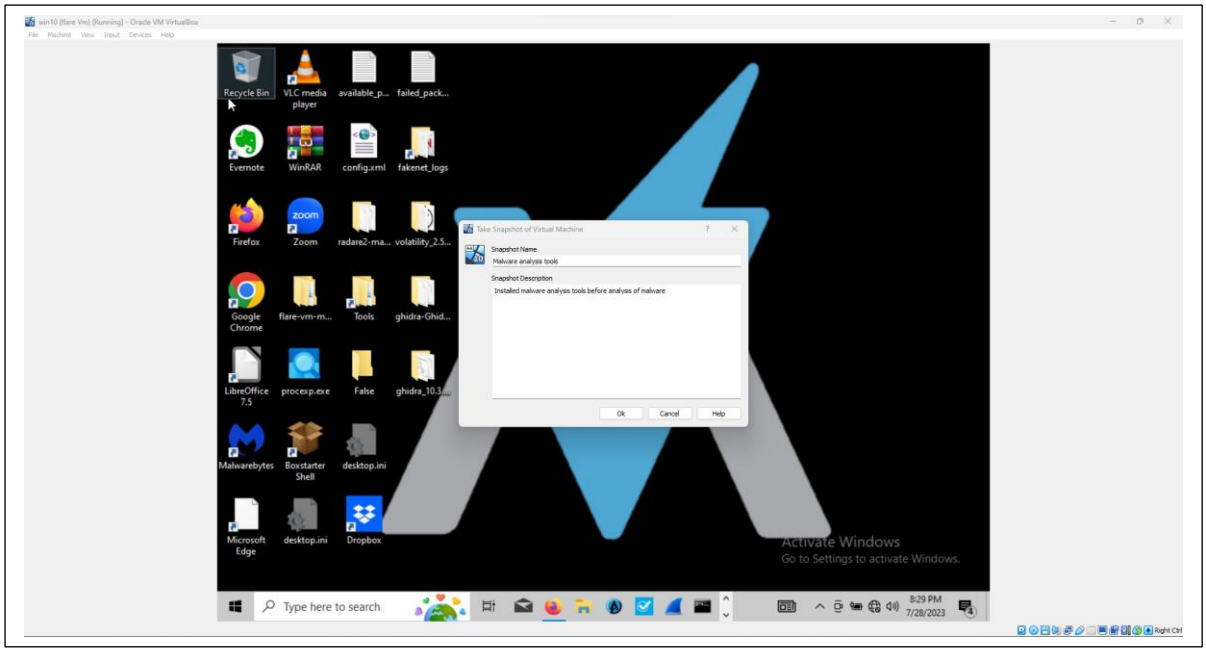

Fig. 8. Malware analysis tools setup

Fig 9 is the Malware analysis tools set-up. Multiple tools were installed using flare-vm (Kacherginsky, 2022)which is an open-source distribution that contains different analysis tools for both forensics and malware analysis.

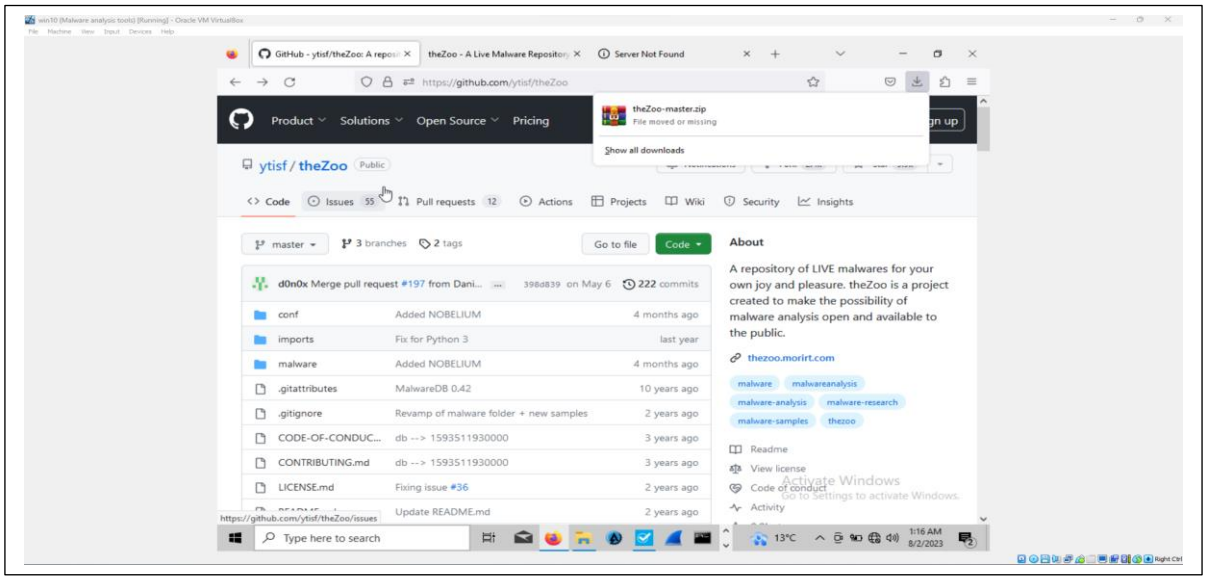

Fig. 9. Malware dataset

The Malware dataset shown in Fig 10 was provided by theZoo -a live malware repository(Nativ and Shalev, no date). The dataset contains a variety of portable executables which are compressed, and password protected. The dataset came with a user manual retrieved from github.com under theZoo's folder. The manual included where to access the malware and its password which is **infected**.

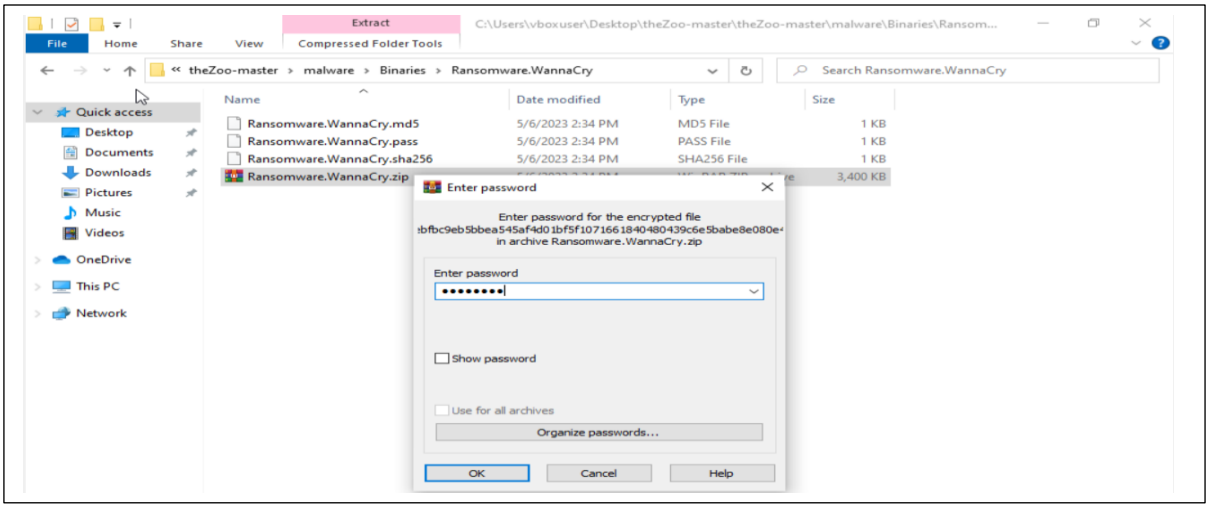

Fig. 10. Ransomware WannaCry

Fig 11 illustrates the WannaCry ransomware. A rule in malware analysis as shown in the figure is portable executables for research are stored in a compressed and password-protected file to limit infecting a device and network.

6 Evaluation

6.1 Case Study 1

6.1.1 Virus total**(virus total, 2023)**

This is an online analysis platform, where we submitted the executables file, and the tool scanned it against multiple antivirus tools and online databases. The tool mainly uses files MD5 and SHA information which is unique to each file. Over 60 security vendors and 5 sandboxes flagged the executable as malicious. The executable was not signed, this failed the signature verification test which is common in malicious files.

Static Analysis	
MD5	84c82835a5d21bbcf75a61706d8ab549
SHA-1	5ff465afaabcbf0750d1a3ab2c2e74f3a44264667
SHA-256	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
File type	Win32 EXE | executable |windows | win32 | pe| peexe
File size	3.35 (3514368 bytes)
PeiD packer	Microsoft Visual C++
Names	tasksche.exe | superkeypass.exe | diskpart.exe | output.251872394.txt
Target machine	Intel 386 or later processors and compatible processors.
History	
Creation time	2010-11-20 | 09:05:05 UTC
First seen	2016-05-16 | 15:27:03 UTC
First submission	2017-05-12 | 07:31:10 UTC
Last analysis	2023-08-02 | 17:00:56 UTC

Table 4. Virus total Static analysis

6.1.2 PEStudio**(Ochsenmeier, 2023)**

Like virus total, Pestudio provides the file's detailed information ranging from SHA & MD5 values which are shown in Table 5. This tool also provides the file-byte-hex and File-bytes-text. It includes a virus total link providing a score of 65 confirmations out of 69.

Static Analysis	
MD5	84c82835a5d21bbcf75a61706d8ab549
SHA-1	5ff465afaabcbf0750d1a3ab2c2e74f3a44264667
SHA-256	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
File-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00
File-bytes-text	MZ „ „ „ „ „ „ „ „
Signature	Microsoft Visual C++ v6.0
File type	executable
File size	3.35 (3514368 bytes)
CPU	32-bit
Compiler-stamp	Sat Nov 20 09:05:05 2010
Subsystem	GUI
Property	
Dynamic-link-library	false
32-bit words support	true
File-executable	true
Machine	Inter-386
Virus total	Score (65/59)

Table 5. PEstudio Static analysis

6.2 Case Study 2

6.2.1 Process Monitor(Russinovich, 2023)

This tool analysed the portable executable by observing file and directory operations, checking if it creates, reads, writes, deletes or/and changes file permissions. The tool also monitors the system's registry, file, network, process, and thread activity. A useful attribute of this tool is creating a process tree, where an analyst can observe the executable's behaviour and any child process created from it. The practical analysis shows that the executable mainly focused on creating new files and changing registry keys and values.

6.2.2 Regshot(TiANWEi, 2023)

This tool takes a file and registry system snapshot before and after executing the portable executable.

Attributes	***1st snapshot (Pre-execution)***	***2nd snapshot (post-execution)***
Date/time	2023-07-28 19:49:13	2023-07-28 19:53:25
Computer	WIN10	WIN10
Username	vboxuser	vboxuser
Keys	463637	463638
Values	773103	773106
Dirs	0	0
Files	0	0
	Analysis	
Keys added	1	
Values added	3	
Values modified	43	

Table 6. Regshot snapshot analysis

6.3 Case Study 3

6.3.1 Ghidra

Ghidra is an extensive reverse engineering tool which performed static and dynamic analysis on the portable executable we provided. It is important to note that the tool focuses on the executable's code to better understand the inner workings of the executable.

Analysis	
Language ID	X86: LE:32:default (2.14)
Compile ID	Windows
Processor	X86
Address size	32
Number of Bytes	3514368
Number of Memory blocks	5
Number of Defined data	542
Number of functions	58
Number of symbols	122
Number of data types	45
Number of data type categories	3
Compiler	visual studio

Executable format	Portable Executable (PE)
MD5	84c82835a5d21bbcf75a61706d8ab549
SHA-256	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
PE property (Company Name)	Microsoft Corporation
PE property (File Description)	Disk part
PE property (Origin filename)	Diskpart.exe
PE property (Product version)	6.1.7601.17514

Table 7. Ghidra analysis

6.4 Discussion

Tools	*Deliverables*				
	User/ Community Support	**Features**	**Usability**	**Effectiveness**	**Cost**
Virus Total	The platform gets frequent support due to its online presence and use of other antivirus software.	Provides file and URL scanning. An analyst can search file hashes and IP addresses. Shares result with the security community.	Basic computing and networking knowledge are needed to upload files and URLs.	The tool compares the executable against previous searches, antivirus and their malware database making it a dependable tool for malware analysis.	Free. Premium services available(*VirusTotal Premium Services*, no date).
PEStudio	The tool provides references, analysis articles and contact information (info@winitor.com)	Provides information ranging from sha-values,bytes-hex, and file-version. Divided into categories; virus total, dos, rich, optional-header, directories, sting, and version.	Intermediate computing, file and operating system knowledge is required to scan and analyse an executable.	The tool provides in-depth information on an executable. Mainly indicator information and dynamic-link-library (DLLs)	Free. Premium services available(*Winitor*, no date).
Process Monitor	This tool has two main supports: Command line options and a help tab that provides more information on the tool and its components.	The tool's main features are the process tree and process, file, registry and network activity analysis	Intermediate computing, file and operating system knowledge is required to scan and analyse an executable.	The tool's process tree enables quick and interpretable data on processes conducted from the executable.	Free.

Regshot	This is a straightforward, easy-to-use tool. Community support provided by Regshot Team via sourceforge.net/projects/regshot/	The tool takes two snapshots of the system and compares the two.	Intermediate computing, file and operating system knowledge is required to scan and analyse an executable	Mainly focuses on the system's files and values.	Free.
Ghidra	The tool provides extensive information on its contents and how to use them.	Conducts static analysis. Mainly focuses on code analysis and reverse engineering.	Expert code analysis skills are required to use the tool	Provides vast code information and a platform to analyse different attributes of the executable's code.	free

Table 8. Comparative analysis

7 Discussion

This research focuses on providing malware analysts a comparative analysis of different malware investigative tools. From this analysis, an investigator is equipped with the right approach when examining malicious software. As investigated, different analysis tools have different functionalities and compatibility. For example, Virus total does not require installation time as compared to tools such as Ghidra or Process monitor. Other tools require more expert knowledge in different fields of computing ranging from coding to operating system.

The main limitations incurred while conducting this research where:

- Cost: This research mainly focused on open-source malware analysis tools (free-readily available tools) due to their availability as opposed to proprietary analysis tools which required payment to be accessed.
- Operating system compatibility: Some of the analysed tools where not compatible with the Windows operating system used for this research.
- Time: Some of the malware analysis tools used for this research required more time to install and configure. This led to spending more time on a specific tool rather than equally on all tools used.

8 Conclusion and Future Work

The research provided a detailed comparative analysis of malware investigative tools focusing on static, dynamic, code and memory analysis. Included is a methodological process of malware analysis using virus total, Promonitor, PEstudio, Regshot and Ghidra analysis tools. The tools analysed WannaCry ransomware and were able to flag it as malicious. Each tool had a specific area of the analysis it focused on from creating a SHA-value to code examination. This research will enable an analyst of either malware, cyber or forensics to in-depth information on how to choose a malware investigative tool efficiently.

Future work may focus on the use of more malware samples and analysis tools that will increase the comparative analysis criteria focusing mainly on a tool's efficiency metrics i.e., true positive, false positive, true negative, false negative, accuracy and detection rate. Different unanalysed malware samples may give more accurate data on the efficiency of the analysis tool.

References

Afianian, A. *et al.* (2020) 'Malware Dynamic Analysis Evasion Techniques: A Survey', *ACM Computing Surveys*, 52(6), pp. 1–28. Available at: https://doi.org/10.1145/3365001.

Aslan, O. and Samet, R. (2017) 'Investigation of Possibilities to Detect Malware Using Existing Tools', in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, Hammamet: IEEE, pp. 1277–1284. Available at: https://doi.org/10.1109/AICCSA.2017.24.

Balodi, B. *et al.* (2023) 'Automated Static Malware Analysis Using Machine Learning', in *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN). 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 255–258. Available at: https://doi.org/10.1109/SPIN57001.2023.10116580.

Barker, D. (2021) *Malware Analysis Techniques*. Packt Publishing. Available at: https://learning.oreilly.com/library/view/malware-analysis-techniques/9781839212277/ (Accessed: 10 August 2023).

Dener, M., Ok, G. and Orman, A. (2022) 'Malware Detection Using Memory Analysis Data in Big Data Environment', *Applied Sciences*, 12(17), p. 8604. Available at: https://doi.org/10.3390/app12178604.

Egele, M. *et al.* (2008) 'A survey on automated dynamic malware-analysis techniques and tools', *ACM Comput. Surv.*, 44(2), p. 6:1-6:42. Available at: https://doi.org/10.1145/2089125.2089126.

Fortinet (2023) *What is Malware Analysis? Types and Stages of Malware Analysis, Fortinet*. Available at: https://www.fortinet.com/resources/cyberglossary/malware-analysis (Accessed: 11 August 2023).

Ghidra (no date). Available at: https://ghidra-sre.org/ (Accessed: 12 August 2023).

'google/grr' (2023). Google. Available at: https://github.com/google/grr (Accessed: 12 August 2023).

Hampton, N., Baig, Z. and Zeadally, S. (2018) 'Ransomware behavioural analysis on windows platforms', *Journal of Information Security and Applications*, 40, pp. 44–51. Available at: https://doi.org/10.1016/j.jisa.2018.02.008.

Ilić, S.Ž. *et al.* (2022) 'A pilot comparative analysis of the Cuckoo and Drakvuf sandboxes: An end-user perspective', *Vojnotehnički glasnik*, 70(2), pp. 372–392. Available at: https://doi.org/10.5937/vojtehg70-36196.

KA, M. (2018) *Learning Malware Analysis*. Packt Publishing. Available at: https://learning.oreilly.com/library/view/learning-malware-analysis/9781788392501/ (Accessed: 21 June 2023).

Kacherginsky, P. (2022) *FLARE VM: The Windows Malware Analysis Distribution You've Always Needed!*, *Mandiant*. Available at: https://www.mandiant.com/resources/blog/flare-vm-the-windows-malware (Accessed: 13 August 2023).

Kayani, A.K. and Saeed, M.Q. (2021) 'Comparative Analysis of Anti-Virus Evasion Malware Creator Tools of Kali Linux, with Proposed Model for Obfuscation', in *2021 International Conference on Cyber Warfare and Security (ICCWS)*. *2021 International Conference on Cyber Warfare and Security (ICCWS)*, pp. 24–29. Available at: https://doi.org/10.1109/ICCWS53234.2021.9702944.

Lebbie, M., Prabhu, S.R. and Agrawal, A.K. (2022) 'Comparative Analysis of Dynamic Malware Analysis Tools', in M. Dua et al. (eds) *Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences*. Singapore: Springer (Algorithms for Intelligent Systems), pp. 359–368. Available at: https://doi.org/10.1007/978-981-16-5747-4_31.

Madan, S., Sofat, S. and Bansal, D. (2022) 'Tools and Techniques for Collection and Analysis of Internet-of-Things malware: A systematic state-of-art review', *Journal of King Saud University - Computer and Information Sciences*, 34(10), pp. 9867–9888. Available at: https://doi.org/10.1016/j.jksuci.2021.12.016.

Maniriho, P., Mahmood, A.N. and Chowdhury, M.J.M. (2022) 'A study on malicious software behaviour analysis and detection techniques: Taxonomy, current trends and challenges', *Future Generation Computer Systems*, 130, pp. 1–18. Available at: https://doi.org/10.1016/j.future.2021.11.030.

Mohanta, A. and Saldanha, A. (no date) *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. Apress. Available at: https://learning.oreilly.com/library/view/malware-analysis-and/9781484261934/ (Accessed: 14 April 2023).

Nativ, Y. and Shalev, S. (no date) *theZoo - A Live Malware Repository*, *theZoo aka Malware DB*. Available at: https://thezoo.morirt.com/ (Accessed: 9 August 2023).

Ochsenmeier, M. (2023) *Winitor*. Available at: https://www.winitor.com/download (Accessed: 12 August 2023).

Oracle VM VirtualBox (no date). Available at: https://www.virtualbox.org/ (Accessed: 12 August 2023).

Preeti and Agrawal, A.K. (2022) 'A Comparative Analysis of Open Source Automated Malware Tools', in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 226–230. Available at: https://doi.org/10.23919/INDIACom54597.2022.9763227.

radare (no date). Available at: https://www.radare.org/n/ (Accessed: 12 August 2023).

Reevell, P. (2021) *Ireland's health service hit by 'significant' ransomware attack, ABC News*. Available at: https://abcnews.go.com/International/irelands-health-service-hit-significant-ransomware-attack/story?id=77685241 (Accessed: 21 June 2023).

Russinovich, M. (2023) *Process Monitor - Sysinternals*. Available at: https://learn.microsoft.com/en-us/sysinternals/downloads/procmon (Accessed: 12 August 2023).

Shijo, P.V. and Salim, A. (2015) 'Integrated Static and Dynamic Analysis for Malware Detection', *Procedia Computer Science*, 46, pp. 804–811. Available at: https://doi.org/10.1016/j.procs.2015.02.149.

Sihwail, R., Omar, K. and Ariffin, K.Z. (2018) 'A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis', *Int. J. Adv. Sci. Eng. Inf. Technol*, 8(4–2), pp. 1662–1671.

Swieskowski, P. and Kuzins, S. (no date) *Ninite - Install or Update Multiple Apps at Once*. Available at: https://ninite.com/ (Accessed: 13 August 2023).

TiANWEi (2023) 'Seabreg/Regshot'. Seabreg. Available at: https://github.com/Seabreg/Regshot (Accessed: 13 August 2023).

virus total (2023) *VirusTotal - Home*. Available at: https://www.virustotal.com/gui/home/upload (Accessed: 12 August 2023).

VirusTotal Premium Services (no date) *VirusTotal*. Available at: https://support.virustotal.com/hc/en-us/articles/115003886005-VirusTotal-Premium-Services (Accessed: 13 August 2023).

Volatility Foundation (2020) *The Volatility Foundation - Open Source Memory Forensics*, *volatilityfoundation*. Available at: https://www.volatilityfoundation.org (Accessed: 11 August 2023).

Whiteman, H. (2022) *Australia blames cyber criminals in Russia for Medibank data breach | CNN Business*, *CNN*. Available at: https://www.cnn.com/2022/11/11/tech/medibank-australia-ransomware-attack-intl-hnk/index.html (Accessed: 9 August 2023).

Winitor (no date). Available at: https://www.winitor.com/pro (Accessed: 13 August 2023).

Yousuf, M.I. *et al.* (2023) 'Windows malware detection based on static analysis with multiple features', *PeerJ Computer Science*, 9, p. e1319. Available at: https://doi.org/10.7717/peerj-cs.1319.